DOCUMENT RESUME

ED 083 814 EM 011 535

AUTHOR Gomberg, Sara
TITLE The REL Command Language.
INSTITUTION California Inst. of Tech., Pasadena.
SPONS AGENCY National Science Foundation, Washington, D.C.; Office of Naval Research, Washington, D.C.; Rome Air Development Center, Griffiss AFB, N.Y.
REPORT NO CAT-REL-R-8
PUB DATE Jun 73
NOTE 16p.; See also EM 011 530 through EM 011 534 and EM 011 536

EDRS PRICE MF-$0.65 HC-$3.29
DESCRIPTORS *Computer Programs; *Computers; *Computer Science; Program Descriptions; *Programing; *Programing Languages
IDENTIFIERS Command Language; Language Implementor Commands; *Rapidly Extensible Language; REL; REL Batch System; Session; Version

ABSTRACT

Three basic notions--version, session and command language--are discussed which are essential for a user of the Rapidly Extensible Language (REL) system. The first is defined as a language--data base package consisting of a language, data and definitions. A session is stipulated to be the period during which the user operates at a terminal, and the command language is the means by which he communicates with and manipulates other versions. Details are next provided about the commands used to initiate and terminate sessions and to manipulate versions. Following this, the enterable and copyable properties of versions are discussed and three categories of users identified--anyone, creator, and no one. Commands used to establish valid user identification names are presented and, lastly, some language implementor commands are described. Examples from the REL batch system are appended. (PB)

a project report on

REL

Co-principal investigators:
Bozena Henisz Dostert
Frederick B. Thompson

California Institute of Technology
Pasadena, California, 91109

The REL Command Language

Sara Gomberg

REL Report No. 8

California Institute of Technology
Pasadena, California

The research reported here was supported in part by the grants:

National Science Foundation grant #GH-31573
Rome Air Development Center contract #F30602-72-C-0249
Office of Naval Research contract #N00014-67-A-0094-0024

INTRODUCTION

Some basic notions with which an REL user must be familiar are: version, session, and command language.

A version is a language-data base package. It consists of a language, data, and definitions. A version will incorporate the grammar and processing routines of the particular language, and the data that the user needs to manipulate via that language. Once a version has been CREATED, it remains in existence until DELETED. (The DELETE and CREATE commands are explained in this paper.) This permits a version to be copied, used, extended, and modified many times. Versions, unless deleted, are automatically retained from session to session.

Versions may be created from "scratch", i.e., a user may program his own language on which he will base his version. More likely, a user will choose to create a version by copying some already existing version. He may then add data and vocabulary or modify his copy as he pleases. The version which is copied is referred to as a base language. For example, RELENGLISH is often used as a base language. It is copied, for example, to create the version TONGA. TONGA consists of a language (RELENGLISH), data about South African tribes, and relevant definitions. A user may want to create a more specialized version than TONGA, but

leave TONGA unaltered. Thus, TONGA would become the base language for this new version.

A session, basically, is the period during which a user is operating at a terminal. A session is initialized by hitting the attention key on the terminal keyboard. During a session a user may create, use, modify, and delete his versions, may enter a number of different versions, and perform certain system functions.

The command language provides a means for the user to communicate with and to manipulate other versions. It is in the command language that the user first communicates with the system. At the moment the user hits the attention key on the terminal, he is in the command language. At this point the user must identify himself to the system. The command language will respond only to a valid user identification. [The authorization of an identification name is discussed later in this paper.] Once a user has identified himself to the command language, he may use the command language to create, enter, copy, or delete a version. He may also put certain controls on the use of a version. The command language also offers some system functions.

The command language command EXIT terminates a session. At this point the terminal becomes available for another user. A user may exit one version and enter another, however, without

terminating a session since each language has its own exit command, which returns the user to the command language.

COMMANDS

Initiating and Terminating a Session

<identification>

The first input to an REL session is an identification. This is any name (20 characters or less) which has been authorized, prior to its use. The user must make arrangements with the proper authority concerning identification before using the REL system.

EXIT

This command terminates a session. In effect, it exits the user from the command language. If one wished to continue using the terminal after entering the command language exit command, he would have to initiate a new session.

Manipulation of Versions

CREATE <version name>

This command enables the user to initially create a version, <version name>, on the basis of a new language. It tells the system to recognize <version name> as a version with an empty language - that is, it has, as yet, no grammar or processing routines. The

create command allows a user to talk to the system about his version, <version name>.

COPY <version name2> FROM <version name1>

A user may want to copy an already existing version, <version name1>, so that he can use or alter it independently of the original. Note, that when he copies a version under a new name, he is considered the creator of this new version, <version name2>. He may give the new version any name he pleases up to 20 characters. He cannot assign a name to his new version that has already been used for some other existent version.

ENTER <version name>

In order to use an existing version, a user must "enter" the version. This is accomplished by the ENTER command. Any statements following this command are interpreted in the language of <version name>, until the user exits from the version, <version name>, back into the command language.

DELETE <version name>

This command completely erases the version specified. The disk space occupied by the version is released back to the system. Only the creator of a version (and the system administrator) can delete a version.

Properties of Versions

Every version has properties which determine who can enter or copy the version. These are set by the creator of the version. For this purpose we have separated users into three categories: 1) ANYONE, 2) ME (creator), and 3) NO ONE. The following six commands control who can copy or enter a version. A version which has just been created can be entered or copied only by its creator until further "marked."

MARK <version name> COPYABLE BY ANYONE

When a version is so marked, any valid user may make a copy.

MARK <version name> ENTERABLE BY ANYONE

A version marked "enterable by anyone" implies that any valid user may enter the version and make changes.

MARK <version name> COPYABLE BY ME

A version so marked could only be copied by the creator, the user under whose I. D. the version was created. For example, a user may decide to mark a version COPYABLE BY ME if the version is still in the "experimental" or "developmental" stages and he does not want any other user to have a copy of it yet.

MARK <version name> ENTERABLE BY ME

In this case, only the creator of the version may enter the version. For example, if a user doesn't wish to have anyone else

enter and perhaps make changes in his version he could use this command to enforce that decision.

MARK <version name> COPYABLE BY NO ONE

When a version is marked copyable by no one, not even the creator of the version may make a copy of his version. A user may wish to mark his version with this command, if he wants only his original version to exist.

MARK <version name> ENTERABLE BY NO ONE

Once a version is so marked, not even the creator may enter the version. A user may use this command to prevent a version from being modified. In this way the version may serve, in a uniform way, as a basis for other versions. An example of this is RELENGLISH. RELENGLISH is a non-enterable version. Another example of use of this property is by an instructor of a class who wishes his students to copy, but not alter, a particular version.

Note that a version's enterable/copyable properties are independent, and thus there are nine possible combinations. Further, only the version's creator may MARK or DELETE that version.

SYSTEM FUNCTIONS

Establishing Valid User Identification

The following two commands, AUTHORIZE and DEAUTHORIZE, are used to establish valid user identification names. These commands are only permitted to a user identified by the special system I. D. RELSYS. RELSYS is pre-authorized at system initialization and should be used with discretion. It has the following special privileges:

1. AUTHORIZE and DEAUTHORIZE are permitted commands.
2. Any version created by any user may be deleted.
3. DIRECTORY lists all versions created by any user.

AUTHORIZE <user id> where <user id> is any identification name of 20 characters or less. A blank is counted as one character.

DEAUTHORIZE <user id> is the command used to delete an identification name from the list of authorized I. D. 's. The deleted I. D. will no longer be recognized by the command language as valid. It may be reauthorized via the authorize command.

DIRECTORY

This command may be used in two ways. First, an authorized user may log-in under his I. D. and type DIRECTORY to get a list of all versions created under his I. D. and their properties. Second, if the user identification is RELSYS then DIRECTORY produces a list of all versions and their respective creators and status.

Language Implementor Commands

SET <version name> GLOBAL AREA SIZE TO <number of bytes>

The processing routines of any REL language have available to them a region called the GLOBAL AREA which may be used at the language implementor's discretion. It is designed, however, for global, but session limited, communications among the routines. Its size is set by this command, and any copies of this version will retain the same size. A version created by the CREATE command has a GLOBAL AREA size of 32.

SET <version name> NUMBER OF EXTENSION PAGES TO <number>

Language routines also have the use of a temporary workspace in the form of a stack. The normal size of this area is approximately one-half page [a page is currently 2048 bytes] but languages whose routines make considerable use of this area may require more space. An indication of this is a "STACK OVERFLOW" abend. This command extends the stack space available by the specified number of pages; the number is retained in all copies of this version. The language implementor should determine the language's needs and set this parameter properly.

EXAMPLES FROM THE REL BATCH SYSTEM

```
SCG
 REL COMMAND LANGUAGE. PROCEED.

COPY SMITH1 FROM SDATA
 VERSION CREATED

ENTER SMITH1
 PROCEED.

WHO ARE THE CHILDREN OF MARY MARTIN?
JACK JONES
JILL JONES SMITH

WHO ARE THE PARENTS OF JACK JONES?
JOHN JONES
MARY MARTIN JONES

WHAT WAS THE INCOME OF JOHN JONES AFTER 1950?
    8000    TO DECEMBER 31, 1952
    9000    FROM JANUARY 1, 1953 TO DECEMBER 31, 1955
   10000    FROM JANUARY 1, 1956 TO DECEMBER 31, 1960
   11000    FROM JANUARY 1, 1961 TO DECEMBER 31, 1969

WHAT WERE THE LOCATIONS OF MARY MARTIN?
BOSTON    FROM DECEMBER 31, 1930

EXIT
RETURNED TO COMMAND LANGUAGE

EXIT
THANK YOU
```

```
SCG
 REL COMMAND LANGUAGE. PROCEED.

COPY VERSION1 FROM SDATA
 VERSION CREATED

MARK VERSION1 ENTERABLE BY ME
VERSION HAS BEEN MARKED

MARK VERSION1 COPYABLE BY NO ONE
VERSION HAS BEEN MARKED

ENTER VERSIUN1
 PROCEED.

DEF:FATHER-IN-LAW:FATHER OF SPOUSE
 DEFINED.

WHO IS THE FATHER-IN-LAW OF JILL JONES?
SAM SMITH

EXIT
RETURNED TO COMMAND LANGUAGE

DELETE VERSION1
 VERSION DELETED

EXIT
THANK YOU
```

```
SCG
 REL COMMAND LANGUAGE. PROCEED.

DIRECTORY
VERSION NAME          COPYABLE   ENTERABLE
SMITH1                 CREATOR    CREATOR
SDATA                  ANYONE     ANYONE

COPY SMITH2 FROM SMITH1
 VERSION CREATED

MARK SMITH1 COPYABLE BY ANYONE
VERSION HAS BEEN MARKED

MARK SMITH2 ENTERABLE BY NO ONE
VERSION HAS BEEN MARKED

DIRECTORY
VERSION NAME          COPYABLE   ENTERABLE
SMITH1                 ANYONE     CREATOR
SMITH2                 CREATOR    NO-ONE
SDATA                  ANYONE     ANYONE

EXIT
THANK YOU
```

```
SCG
 REL COMMAND LANGUAGE. PROCEED.

CREATE SMITH3
 VERSION CREATED

EXIT
THANK YOU
```

```
RELSYS
 REL COMMAND LANGUAGE. PROCEED.

AUTHORIZE LDG
ID IS AUTHORIZED

DIRECTORY
REL SYSTEM VERSION DUMP
CREATOR NAME        VERSION NAME        COPYABLE    ENTERABLE
FBT
                    ENGSYN              ANYONE      CREATOR
                    TDATA               ANYONE      CREATOR
                    TD1                 CREATOR     CREATOR
LEN
                    LENTST              CREATOR     CREATOR
SCG
                    SMITH1              ANYONE      CREATOR
                    SMITH3              CREATOR     CREATOR
                    SMITH2              CREATOR     NO-ONE
                    SDATA               ANYONE      ANYONE
PMN
                    STATISTC            ANYONE      ANYONE

DELETE SMITH1
 VERSION DELETED

DELETE SMITH2
 VERSION DELETED

DELETE SMITH3
 VERSION DELETED

EXIT
THANK YOU
```